**AFRL-IF-RS-TR-2007-132**
**Final Technical Report**
**May 2007**

# AUTOMATIC DETECTION OF COVERT CHANNELS IN NETWORKS

**Tufts University**

*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*

**STINFO COPY**

**AIR FORCE RESEARCH LABORATORY**
**INFORMATION DIRECTORATE**
**ROME RESEARCH SITE**
**ROME, NEW YORK**

# NOTICE AND SIGNATURE PAGE



AFRL-IF-RS-TR-2007-132 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/ /s/

ANDREW KARAM
Work Unit Manager

WARREN H. DEBANY, Jr.
Technical Advisor, Information Grid Division
Information Directorate

# REPORT DOCUMENTATION PAGE

*Form Approved*
**OMB No. 0704-0188**



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| MAY 2007 | Final | Nov 05 – Mar 07 |

**4. TITLE AND SUBTITLE**
AUTOMATIC DETECTION OF COVERT CHANNELS IN NETWORKS

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**
FA8750-05-2-0015

**5c. PROGRAM ELEMENT NUMBER**
62702F

**6. AUTHOR(S)**
C.E. Brodley

**5d. PROJECT NUMBER**
4519

**5e. TASK NUMBER**
TB

**5f. WORK UNIT NUMBER**
03

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Tufts University
28 Sawyer Ave
Medford MA 02155-5581

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
AFRL/IFGB
525 Brooks Rd
Rome NY 13441-4505

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSORING/MONITORING AGENCY REPORT NUMBER**
AFRL-IF-RS-TR-2007-132

**12. DISTRIBUTION AVAILABILITY STATEMENT**
*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 07-230*

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**
A covert channel is a mechanism that can be used to violate a security policy by allowing information to leak to an unauthorized process. Two types of covert channels exist; storage and timing channels. A storage channel involves the direct or indirect writing of a storage location by one process and the direct or indirect reading of the storage by another process. A timing channel involves a sender process that signals information to another by modulating its own use of systems resources in such a way that this manipulation affects the real response time observed by the second process. In this research, we focused on the analysis and detection of covert timing channels in the TCP/IP protocol suite.

**15. SUBJECT TERMS**
Covert timing channels, IP covert timing channels, Detection of Stack Buffer Overflow Attacks

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Andrew J. Karam |
| U | U | U | UL | 14 | 19b. TELEPHONE NUMBER (*Include area code)* |

# TABLE OF CONTENTS



# LIST OF FIGURES

## 1. Background

A covert channel is a mechanism that can be used to violate a security policy by allowing information to leak to an unauthorized process [4]. Two types of covert channels exist: storage and timing channels. A storage channel ``involves the direct or indirect writing of a storage location by one process and the direct or indirect reading of the storage location by another process" [9]. A timing channel involves a sender process that ``signals information to another by modulating its own use of system resources (e.g., CPU time) in such a way that this manipulation affects the real response time observed by the second process" [13]. In this research, we focused on the analysis and detection of covert timing channels in the TCP/IP protocol suite. Although some work has been done on timing channel analysis in general, little attention has been paid to channels in IP.

## 2. Research Objectives and Contributions

Our research addressed event-based network covert channels that arise in distributed TCP/IP MLS systems even when the transmission lines between network nodes are controlled. In our model Alice and Eve are not allowed direct communication. Alice is communicating with a third party (Bob) and Eve is able to watch Alice's network traffic. Our research answers to two questions: How can we design network covert channels using the TCP/IP protocol to leak information from Alice to Eve? How can the system detect and eliminate such leakage? In order to provide answers, we investigate methods for designing and countering event-based covert channels. In particular,

1) We designed and implemented a prototype IP covert channel using a novel design in which we use IP packet timings to transmit covert information over the network effectively.

2) We introduced novel detection measures that effectively detect both noiseless and noisy IP covert channels (and all event-based covert channels in general) using traffic analysis and force malicious users to either design more complex channels and/or rate-limit the channel to avoid detection.

3) We introduced time-replay covert channels, which is a covert channel family that hides event-based channels, and demonstrated that TRCCs are virtually undetectable under certain assumptions.

4) We investigated prevention and elimination techniques for time-replay covert channels that aid current elimination schemes in stopping these channels.

## 3. Designing Unhidden Network Covert Channels

Our research initially focused on designing event-based storage and timing IP simple covert channels (SCC). Our design of IP SCCs uses the packet arrival as the event and employs the inter-arrival times between the IP packets to convey information in distributed MLS systems. While simple in concept, there proved to be some non-obvious issues in creating the channel and designing the software. One subtle issue is the synchronization of Alice and Eve's event clocks in order to guarantee channel accuracy. Because IP packets offer no guarantees on the time of packet deliveries, we employed additional schemes to preserve synchronization and resynchronize the channel as needed. In the presence of the factors that introduce noise into the covert channel, we assessed the efficacy of IP SCCs in terms of bit/character accuracy (the number of bits/characters transmitted correctly divided by the total number of bits/characters) and channel bandwidth. To do so, we perform basic accuracy and bandwidth analysis for both noiseless and noisy IP SCCs. We show that two factors affect channel efficacy: *contention noise*, which is the amount of non-covert traffic Eve observes in the covert channel and can potentially reduce channel accuracy, and *clock skew*, which is the amount of jitter in the network and can potentially result in the loss of synchronization between Alice's and Eve's event clocks. The results of these experiments are reported in [11] and [12].

We implemented our covert channel as a client and server using Berkeley sockets library in C for our communication protocol, and Python version 2.3 to encode/decode the data sent on the channel and as a wrapper that called the C library functions. This software was developed for and ran under RedHat Linux 9.0 kernel version 2.4.20.

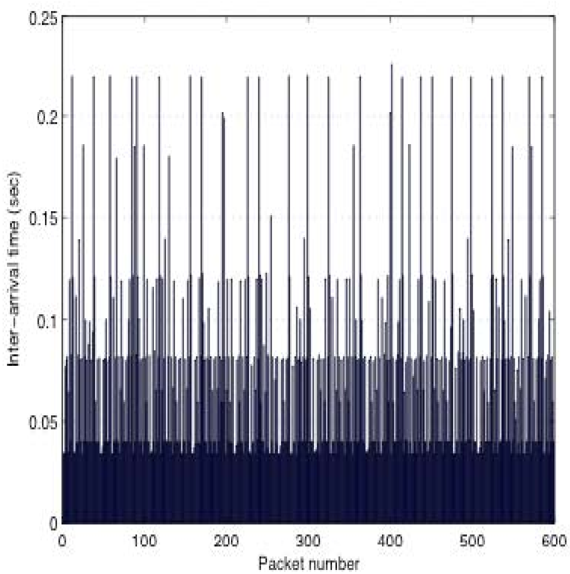
## 4. Countering Unhidden Network Covert Channels

In order to counter these channels, we investigated IP SCC detection, elimination, and rate-limiting techniques. In particular, we developed techniques to detect both noiseless and noisy IP SCCs. Our analysis of the behavior of IP SCCs illustrates that both storage and timing channels show some type of regularity in packet inter-arrival times. This is an expected behavior for storage channels, because they have a fixed timing interval to send/not send the packets. Timing channels, on the other hand, do not use constant timing intervals. However, in a straightforward implementation, a symbol is sent using only a limited number of these timing intervals and these intervals repeat over time. Hence, the packet inter-arrival times for both storage and timing IP SCCs are repetitive. For example, the inter-arrival times for a storage covert channel in Figure 1(a) illustrate this regularity, whereas traces of normal traffic exhibit no such regularity can be deduced. Although, for such legitimate channels, user and network affects can potentially be observed on the traffic pattern on a longer window. However, we claim that, in most cases (e.g., except streaming traffic), legitimate traffic patterns are much less repetitive than IP SCC traffic.

In order to detect this type of regularity, we introduced two detection measures and evaluated their capabilities in an experimental study using our covert channel software in a real-world scenario. Further, we showed that our detection measures fail for noisy covert channels for noise levels higher than 10%. To counter this deficiency, we investigated effective yet computationally expensive search mechanisms to locate the hidden covert channels locally for which the global

measures fail. An important observation here is that this regularity characteristic is not limited solely to the channels we designed but to all event-based covert channels. Hence, our detection measures can be employed not only for IP SCCs but for any channel that is based on event timings. The results of these experiments were presented at the ACM Conference on Communications and Computer Security [11] and have been submitted to TSSEC [12].

In Figure 1(a) we show the inter-arrival times of a simple covert timing channel. The y-axis is the inter-arrival time and the x-axis is the packet number. In Figure 1(b), we have sorted the inter-arrival times from smallest to largest. The result is a step function (note that because of varying network load, it is not a perfect step function). From these two figures, we observe that there appear to be approximately 4 or 5 different inter-arrival times. This highly regular behavior is a direct result of the static encoding of the frames in the timing channel. The arrival of packets is separated by 0, 1, 2, 3, 4,... intervals (the number of intervals separating packets is the number of ``zeros'' between two consecutive ``ones'' in a codeword). In contrast for normal non-covert traffic packets can arrive anytime, resulting in an irregular pattern. Our two preliminary measures are as follows:



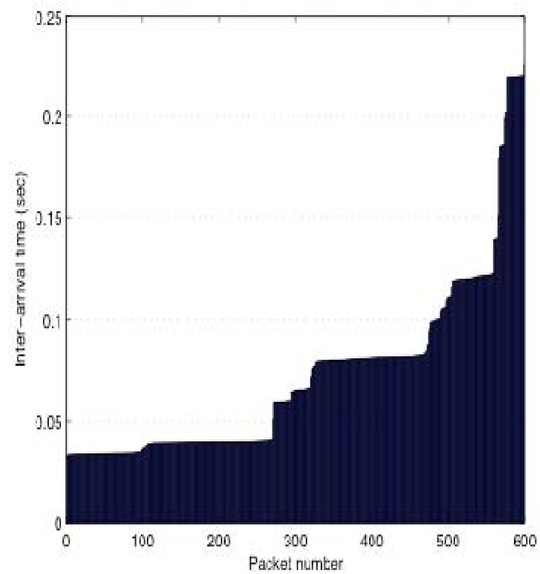




**Figure 1(a): The left-hand-side shows the inter arrival times. (b) The right hand side shows the times sorted.**

Measure 1: Our first method examines whether the variance in the inter-arrival (IA) remains constant. To this end, we separate the traffic into non-overlapping windows of size w packets. For each window i, we compute the standard deviation of the IA times. To compute our heuristic measure of regularity, we then calculate the pairwise differences between the standard deviations for each pair of windows. Finally to obtain a summary statistic, we compute the standard deviation of the pairwise differences.

Measure 2: Our second measure, which we call epsilon-Similarity, is derived from the sorted IA times. From this sorted list, we compute the relative difference between each pair of consecutive points. For example the relative difference between a and b is computed as |a-b|/a. We can then compute a measure of similarity by computing the percentage of relative differences that are less than a specified parameter, epsilon. For covert channels the majority of the pairwise differences in the sorted list of IA times will be very small. It is large only for jumps in the step function.

The goal of our experiments was to examine the efficacy of our two metrics. To this end, we experimented with the basic covert timing channel and then two channels for which we tried to make the channel more difficult to detect. In our experiments, we employ the second version of NZIX data sets (NZIX-II) which is a collection of TCP and UDP traces collected by the WAND research group [6]. For the TCP traces, we chose to investigate Telnet, FTP, and HTTP traffic. For each experiment we report results for traffic flows of 2000 packets. Our goal is not to model or identify a traffic distribution, but to determine whether we can accurately detect a covert channel in a window of 2000 packets.. Note that although the covert channel was run between Purdue and Georgetown Universities, for the non-covert traffic we use the recorded IA times in the datasets. A drawback is that we cannot have the same network conditions (e.g., number of hops, same jitter), but excluding the case of jitter this does not impact our results. None of our measures look at absolute IA values, but rather compute measures of regularity in terms of the relative differences among IA values. For our simple channel, both metrics work perfectly with 100 percent detection accuracy and no false alarms. When noise is introduced into the channel by interjecting normal traffic at fixed intervals our first metric fails to detect the covert traffic, However our second metric performs well until noise levels reach above 25-50% Details of these results can be found in our CCS paper.

## 5. Designing Hidden Network Covert Channels

Despite the characteristic regularity of IP SCCs and event-based channel channels in general, more sophisticated schemes that hide the channel better can be devised. In the second part of our research, we developed Time-Replay Covert Channels (TRCC) which are specifically designed to hide covert traffic by adjusting packet timings consistent with inter-arrival time sequences that are extracts from recently recorded normal sequences under similar network conditions. Such time-replay channels represent a family of covert channels rather than a particular channel or a channel type.

Time-replay covert channels replay a previously recorded event activity using a simple but an effective three-fold strategy: 1) obtain a pre-recorded event sequence as input, 2) divide the sequence into ***k*** partitions, where ***k*** is the size of the message alphabet, and associate each partition with a symbol using a set of thresholds (i.e., rules), and 3) send the symbol ***s*** by delaying for the amount of time indicated by the timing value in the corresponding partition before generating the event. For example, suppose that a malicious user Alice wishes to leak code ***C*** to an eavesdropper Eve using a time-replay covert channel. Suppose ***C*** is made up from two types of symbols: ***s_1*** and ***s_2*** (e.g., *zero* and *one* for binary codes). To send ***s_1***, Alice chooses a timing value ***T*** from the ***s_1***-partition of the input sequence and generates an event after idling for ***T*** time. To send ***s_2***, she uses the ***s_2***-partition. Each timing value in each

partition is used only once. On the receiving side, Eve monitors the events. Upon observing an event, Eve 1) calculates the inter-event time ***T*** between the current and the most recent event, 2) determines to which partition ***T*** belongs, and then 3) records ***s_1*** or ***s_2*** depending on the decision in step 2. This is an example of a *binary matching channel*.

We illustrated that using a simple IP channel, called the binary-matching channel, Alice and Eve can exchange messages effectively over the network with high accuracy and secrecy. Further, we investigate the channel efficacy of IP TRCCs and assess it in terms of bit/character accuracy (the number of bits/characters transmitted correctly divided by the total number of bits/characters) and channel bandwidth.

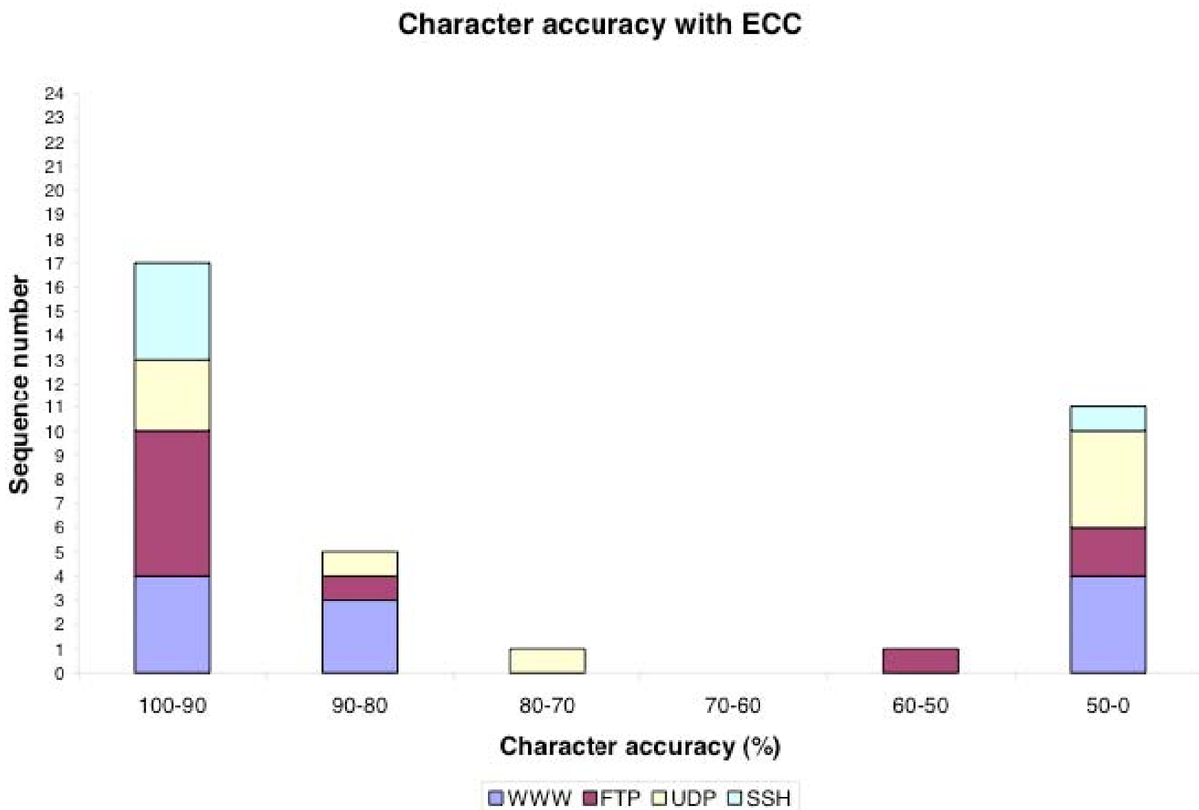



**Figure 2: We show the character accuracy as computed by the edit distance from the captured bits. Sequences are split into different traffic types and flows.**

In order to get an upper-bound on channel accuracy, we ran BMC between two machines on the same LAN to send the ASCII-encoding of a simple text. In Figure 2 we report the character accuracy results error-correction for all 35 NZIX-II inter-arrival time sequences using histograms. We observe that 22 sequences yielded over 80% character accuracy rates, 17 of which yielded over 90% accuracy. From these results we observed that in addition to contention noise and clock skew as in IP SCCs, a third factor, *sequence selection* (i.e., which sequence the time-replay covert channel uses) plays a major role in channel accuracy and bandwidth. The goal is to have inter-arrival times that are not clustered around the threshold value(s) for the partitions. The 17 sequences with the highest accuracy were for IA sequences that were had this

characteristic (see Figure 3).

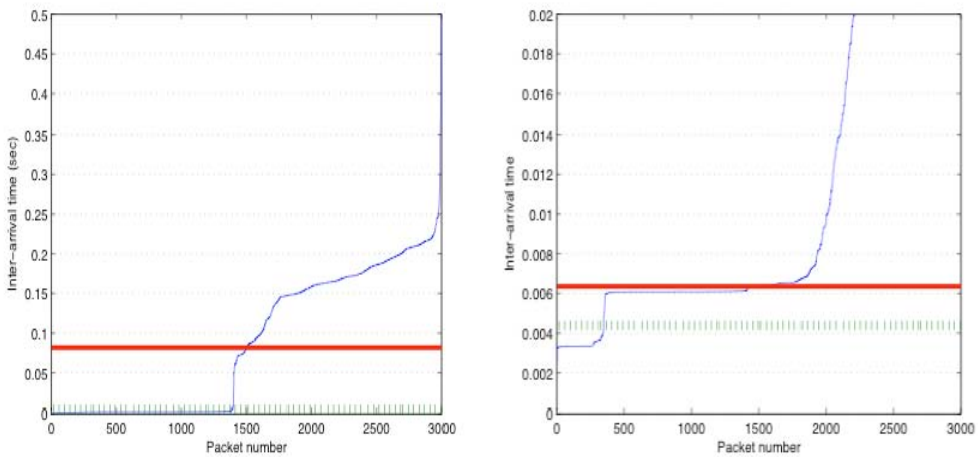



**Figure 3: On the left we show the NZIX-II WWW time sequence that yield 100% character accuracy for BMC. The solid line is the threshold used by the BMC. On the right we show an FTP sequence that yielded very low character accuracy.**

## 6. Countering Hidden Network Covert Channels

In order to counter IP TRCCs, we investigated covert channel detection, elimination, and prevention techniques. We first investigated detection and elimination schemes that are applicable to IP time-replay covert channels and the conditions under which these schemes are effective. Note that because IP TRCCs use replays of inter-arrival sequences extracted from legitimate sources, the traffic pattern shows no regularity as in the case of IP SCCs and the regularity-based detection measures do not apply to these channels. Further, we showed that a sophisticated attacker can potentially devise and utilize covert time sequences that can be made indistinguishable from normal sequences that were generated by legitimate channels, hence creating virtually undetectable replay covert channels. Thus, to deal with these channels, we investigated how the existing covert channel elimination schemes need to be revised to eliminate the new class of covert channels we introduce. In particular, we focused on two covert channel elimination schemes, jamming [3] and the network pump [8], and evaluated their effectiveness on IP TRCCs. We showed that the binary-matching channel may survive these elimination schemes with a slightly lowered data rate. (The actual amount by which the data rate is reduced depends on the amount of added average delay to the packets.) We argue that, in order to eliminate these channels, elimination schemes need to transform packet timings more intelligently in order to remove the associations between the timing values and symbols.

## 7. Significance of the Work

Handling covert channels, that is identifying, analyzing, limiting, and eliminating these channels, is particularly important for multi-level secure (MLS) systems, in which processes may leak

information to other processes with a lower classification level via the use of shared resources [9]. Indeed, the original (but now obsolete and replaced by Common Criteria (CC)) evaluation criteria for trusted computer systems (TCSEC: 1983--1999 --The Orange Book) included requirements to analyze covert channels in terms of their bandwidth and to develop policies to monitor and maintain their bandwidth below maximum acceptable levels [13]. In particular, TCSEC required storage channel analysis for a B2 system, and a complete analysis for B3 and higher assurance level systems. These requirements were also carried to the new version of evaluation criteria: the Common Criteria (CC: 1998--Present). CC requires covert channel analysis for an EAL5 assurance level as well as a systematic search for covert channels for EAL6 and higher assurance levels.

Despite their importance in secure systems, auditing and detecting covert channels have received less attention compared to other covert channel handling techniques. Furthermore, covert channel detection still remains a novel area often confused with covert channel identification (as in [1][5]. The identification problem is to discover the potential covert channels that can be realized in the analyzed system. In contrast, covert channel detection mechanisms are similar to intrusion detection systems. The essential task in covert channel detection is to detect anomalous traffic patterns that can potentially signal the presence of a covert channel. In addition to this, an ideal detection scheme would be the one that identifies the communicating parties, the information transferred, and the channel capacity.

Network covert channels are particularly important when the receiver of the (direct) channel (i.e., Bob) is not in the same system as the sender (i.e., Alice) and the access is controlled between the two. These channels pose a threat to distributed systems in which sensitive information is stored. This is because once an attacker (i.e., Eve) implants a back-door into one of these systems, she can now steal information by leaking it through the covert channel that can be established using the back-door program. Moreover, even without the presence of this back-door, Eve can monitor the traffic between Alice and Bob to gather information Alice is trying to leak. Without these channels, this scenario is not possible because direct access to the untrusted parties (e.g., the nodes outside the trusted network), including Eve, is controlled by policy.

Detection is a common practice in secure systems in order to monitor malicious activity. Detecting covert channels is desirable for three reasons: Firstly, detection provides a mechanism to discourage the use of these channels and may work as a deterrent [2]. Secondly, most covert channel identification systems need input from system analysts (i.e., to specify the shared resources). Due to human error, a number of covert channels may remain unidentified. Detection can help to record the activity of these channels. And lastly, covert channel elimination can be very costly for high performance systems [10]. In this case, allowing these channels to exist but monitoring their activity is crucial.

## 8. Products/Articles from this Research

- Cabuk, S., Brodley, C. E., and Shields, T. C., ``IP covert timing channels: An initial exploration," *The Eleventh ACM Conference on Computer and Communications Security*, Washington D.C., October 25-29, 2004

- Cabuk, S., Brodley, C. E. and Shields, C., ``IP Covert Channel Detection," submitted to *ACM Transactions on Information and Systems Security (TISSEC*).

- Cabuk, S. *Network Covert Channels: Design, Analysis, Detection and Elimination*, Purdue University Ph.D. Thesis, December 2006.

- Cabuk, S, Brodley, C. E., and Spafford, E., "Time Replay Covert Channels," in preparation.

- Implemented covert-communication channel (see above for a description).

## 9. References


[1] Matt Bishop, *Computer Security: Art and Science*, Addison Wesley Professional, 2002

[2] Patrick R. Gallagher, Jr., "A guide to understanding covert channel analysis of trusted systems," *National Computer Security Centre NCSC--TG—030*, Library No. S240,572, 1993

[3] James Giles and Bruce Hajek. An information-theoretic and game-theoretic study of timing channels. In IEEE Transaction on Information Theory, volume 48, pages 2455–2477, September 2003.

[4] Virgil Gligor. A guide to understanding covert channel analysis of trusted systems. Technical Report NCSC-TG-030, National Computer Security Center, Ft. George G. Meade, Maryland, U.S.A., November 1993.

[5] WAND Research group. NZIX-II trace archive, data available at http://pma.nlanr.net/traces/long/nzix2.html.

[6] Loc Helouet, Claude Jard and Marc Zeitoun, Covert channels detection in protocols using scenarios," Proceedings of *SPV'2003, Workshop on Security Protocols Verification*, 2003.

[7] M. Kang, I. Moskowitz, and D. Lee. A network version of the pump. In Proceedings of the IEEE Symposium in Security and Privacy, pages 144–154, May 1995.

[8] Myong H. Kang, Ira S. Moskowitz and Daniel C. Lee, "A Network Pump, *IEEE Transactions on Software Engineering*, 22(5), pp 329-338, 1996.

[9] Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, and Kumar Das. The 1999 DARPA off-line intrusion detection evaluation. Computer Networks, 34(4):579– 595, 2000.

[10] John McHugh. Covert channel analysis. Technical report, December 1995.

[11] Cabuk, S., Brodley, C. E., and Shields, T. C., ``IP covert timing channels: An initial exploration," *The Eleventh ACM Conference on Computer and Communications Security*, Washington D.C., October 25-29, 2004

[12] Cabuk, S., Brodley, C. E. and Shields, C., ``IP Covert Channel Detection," submitted to *ACM Transactions on Information and Systems Security (TISSEC*).

[13] U.S. Department of Defense. Trusted computer system evaluation ”The Orange Book”. DoD 5200.28-STD Washington: GPO:1985, 1985.